

Khai báo biến, scope, hoisting

var

```
var name = 'Duoc'  
console.log(name)  
name = 'Hung'  
console.log(name)
```

Khởi tạo

```
var age  
console.log(age)  
age = 18  
console.log(age)
```

Quy tắc đặt tên

Tên phải bắt đầu bằng chữ cái hoặc ký tự `_` `$`. Không được bắt đầu bằng số

```
var _car = 'BMW'  
var $stream = true
```

Một số quy ước đặt tên

```
var firstName = 'Du' // Camel case  
var first_name = 'Du' // Underscore  
var FirstName = 'Du' // Pascal case
```

Vấn đề hoisting với var

```
console.log(message)  
var message = 'Bị hoisting'
```

tương đương

```
var message  
console.log(message)
```

```
message = 'Bị hoisting'
```

Chế độ bình thường cho phép khởi tạo biến mà không cần dùng var

```
message = 'Thông báo'  
console.log(message) // Thông báo
```

Cách fix với use strict

```
'use strict'  
message = 'Thông báo'  
console.log(message) // Thông báo
```

let tương tự var nhưng không bị hoisting

```
console.log(message) // lỗi  
let message = 'không bị hoisting'
```

const cũng không bị hoisting

const dùng để khai báo hằng số không thể bị gán lại

```
const car = 'Mercedes'  
car = 'BMW' // Lỗi
```

Để ý với object

```
const profile = { name: 'Duoc' }  
profile.name = 'Huy'  
const array = [1, 2, 3, 4]  
array.push(5)
```

scope

Global Scope

```
var company = 'xdevclass'  
// Code tại đây có thể xử dụng biến company  
function myFunction() {
```

```
// Code tại đây cũng có thể sử dụng biến company  
}
```

Function Scope

Code ở đây không thể sử dụng biến carName

```
function myFunction() {  
  var carName = 'Volvo'  
  // code ở đây CÓ THỂ sử dụng biến carName  
}  
// Code ở đây không thể sử dụng biến carName
```

Block Scope

```
{  
  var x = 2  
}  
// x CÓ THỂ được dùng tại đây
```

Trước ES2015 (ES6) Javascript không có **Block Scope**.

Từ ES6 trở đi biến được khai báo bằng **let** và **const** có **Block Scope**. Vì thế các biến bên ngoài block { } không thể được truy cập vào bên trong block

```
{  
  let x = 2  
}  
// x KHÔNG THỂ truy cập tại đây
```